

PATENT
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
(oracle01.027)

5 **Applicant:** Barton, et al. **Confirmation No:** 9521
 Application No: 10/656,525 **Group Art Unit:** 2168
 Filed: 9/3/2003 **Examiner:** Morrison, Jay A.
 10 **Title:** *Apparatus and methods for transferring database objects into and out of
 database systems*

15 Commissioner for Patents
 Alexandria, MA 22313-1450

Appeal Brief under 37 C.F.R. 41.37

20 **(1) Real party in interest**

The real party in interest is the assignee of record of the above application, Oracle International Corporation, which is a wholly-owned subsidiary of Oracle Corporation.

25 **(2) Related appeals and interferences**

None.

30 **(3) Status of claims**

Claims 1, 3-38, and 43-65 are presently in the application. All of these claims have been rejected. Claims 2, 39-42, 52, and 64 have been canceled.

35 **(4) Status of amendments**

A first Amendment After Final was filed 2/18/07 which canceled claims 39-42; Private Pair indicates that the amendment was entered, but the "Notice of Panel Decision from Pre-Appeal Brief Review" which was mailed 3/16/07 indicates that claims 39-42 stand rejected. In preparing this Brief, Applicants noted a number of errors:

- that claim 59 should be dependent from claim 58 instead of claim 1;
- that claim 64, a Beauregard claim dependent from claim 39, should have been canceled when claim 39 was canceled; and
- that claim 17 was misdescriptive of the invention to which it is directed.

5 A second Amendment after Final which corrected the first two of these errors was filed electronically on 4/17/07. A third Amendment after Final which corrected the error regarding claim 17 was filed electronically on 4/18/07. None of these corrections affect the rejections with which this Appeal is concerned; consequently, in the following Brief, the claims will be treated as though the second and third Amendments after Final had
10 been entered.

(5) Summary of claimed subject matter

Overview of the claimed subject matter

15 The claimed subject matter is techniques for transferring database objects between database systems. Transferring database objects between database systems is made difficult by the fact that the objects may be very large and by the fact that any such transfer must maintain the objects' organization, including how the objects are ordered and the relationships between them. Most database systems have *export* and *import*
20 utilities of the types shown in FIGs. 1 and 2 for performing such transfers. The export utility puts the objects into a form which permits another database system to reconstruct the objects and the import utility reconstructs the objects from the exported form. Utilities also exist for importing data into a database from a non-database source.

25 While these utilities do perform their intended functions, they have a number of drawbacks which make them inefficient and hard to use:

- The utilities for exporting objects to and importing objects from other database systems write and read export files 105 serially; consequently, these utilities cannot take advantage of the database system's ability to perform operations in parallel to
30 speed the export and import operations.
- None of the export or import utilities is restartable; if an export or import job is

interrupted at any point for any reason, it must be started over again from the beginning.

- The only status information provided for an ongoing export or import operation is the log file produced by the utility and many events in the operation do not result in an entry in the log file.
- Once an export or import job is started, the operator has no further control over it.
- While the utilities may be used to do fileless transfers, none of them has been specifically designed for that purpose.
- Current export and import utilities have limited capabilities for transforming metadata during an import operation or filtering and selecting objects during either import or export.

The techniques of the invention overcome these drawbacks by employing a control object in the database management system to control performance of a job which transfers a set of objects into or out of the data base management system by a mechanism that transfers database objects. In a preferred embodiment, the control object is a table, the master table, in the database system that is performing the transfer. The control object may specify a set of the objects to be transferred, an order in which the transfer mechanism transfers objects, or a filter that further specifies the set of objects. The control object may further specify an operation on one or more objects being transferred, including a transformation of the object or a remapping of a name in the object. The control object may also specify one or more parameters for the job.

The control object further includes a specification of a status of the job which is updated by the transfer mechanism during the transfer. The control object is queryable to obtain a current status of the job from the specification of the status and the data transfer mechanism uses the specification of the status of the job to restart the job after the job has been stopped.

The claims

There are 59 claims remaining in the application. The independent claims are 1, 36, and

43. Groups of related claims and where they are supported in the Drawing and Specification are indicated in the following table:

Group ID	Claims	Where supported
A	1, 43, 62, 65	Fig. 3, page 7, line 30-page 10, line 2
B	3-4	Fig. 5, Figs. 10-12; page 10, lines 30-34; page 15, lines 15-19
C	5	Fig. 5, Fig. 17; p. 10, lines 14-15; p. 28, lines 1-15
D	6-8	Fig. 5, Fig. 17; p. 10, lines 16-17; page 28, lines 18-24
E	9-11	Fig. 5, Fig. 16; p. 10, lines 8 and 9; page 27, lines 10-18
F	12-13	Fig. 12, Fig. 24; p. 16, line 29-page 17, line 19
G	14-23; 60-61	Fig. 6; page 12, line 20-page 14, line 16
H	24	Fig. 7; page 14, lines 20-23, page 15, lines 21-32
I	25	Fig. 21; page 29, line 10-page 35, line 16
J	26-35	Fig. 6, Fig. 21; page 12, line 20-page 14, line 16; page 29, line 10-page 35, line 16
K	44-60	Fig. 4, Fig. 6; page 19, line 24-page 21, line 17; page 12, line 20-page 14, line 16;
L	36-38, 63	Fig. 3; Fig. 9; page 8, line 31-page 9, line 14; page 11, line 31-page 12, line 17;

- 5 It should be noted here that in the present patent application, the reference numbers have two parts: the rightmost two digits specify a number within the figure in which the item to which the reference number refers first appears and the remaining digits specify that figure. Thus, the item referred to by the reference number "203" will first be found in FIG. 2. Reference numbers employed in the claims in the following discussion are for
10 the convenience of the Board of Appeals and are not intended to limit the claims.

The claims of Group A

- Group A includes two independent claims 1, and 43, and two dependent Beauregard claims whose patentability depends completely on the patentability of the independent
15 claims. These claims are supported at least by FIG. 3, the description at page 7, line 30 through page 10, line 2, and FIG. 5 and the description at page 10, line 5 through page 11, line 19. Claim 1 is addressed to apparatus which embodies the principles of the invention and claim 43 is addressed to a method which embodies the principles of the

invention. Claim 1 is typical; claim 43 has substantially the same limitations and is supported by the same disclosure.

1. Apparatus in a database management system (301) for performing a job which transfers a set of database objects into or out of the database management system, the apparatus comprising:
 - a transfer mechanism (311) that transfers database objects; and
 - a queryable control database object (321) that represents the job and specifies the set of objects (515),
 - the transfer mechanism operating under control of the control database object to transfer the objects in the set.

The "database management system" is shown at 301 in FIG. 3; the transfer mechanism is embodied in datapump 311; the queryable control database object is embodied in master table 311; the "set of objects" is specified by object info 515; a row of master table 321 which represents an object is shown at 507.

The claims of group B

Claims 3 and 4 are dependent from claim 1. They address the mechanisms by which the control database object specifies an order in which the transfer mechanism transfers the objects in the set. As explained at page 10, lines 30-34 and in more detail at page 22, lines 16-22; as set forth there, the order in which objects are processed on import or export depends on the values of fields 503, 508, and 504. That objects are ordered by size is set forth at page 15, lines 15-19.

The claim of group C

Claim 5 is addressed to the mechanism used in the control database object to specify a filter that further determines which objects that are to be included in the set of objects. The filter information is shown in overview at 523 in FIG. 5 and shown in detail at 1701 and 1709 in Fig. 17. A detailed description may be found at page 28, lines 1-15.

The claims of group D

Group D includes claims 6-8, which are dependent from claim 1. These claims are addressed to the mechanism used in the control database object to specify operations that

are to be performed on objects as they are imported or exported. The operation information is shown in overview at 525 in FIG. 5, shown in detail at 1715 in FIG. 17 and described in detail at page 28, lines 18-24.

5 *The claims of group E*

Group E includes claims 9-11, which are dependent from claim 1. These claims are addressed to the mechanism used in the control database object to specify the status of the job. The status information is shown in overview at 517 in FIG. 5, shown in detail at 1601 in FIG. 16, and described in detail at page 27, lines 10-19.

10

The claims of group F

Group F includes claims 12 and 13, which are dependent from claim 1. These claims are addressed to situations in which a remote database is the source of the set of objects. Claim 12 is supported by REMOTE_LINK at 1207 in FIG. 12 and page 15, lines 21-32; claim 13 is addressed to fileless import, which is disclosed in the flowchart of FIG. 24 and the discussion at page 16, line 29-page 17, line 19.

15

The claims of group G

Group G includes claims 14-23 and claims 60-61, which are all dependent from claim 1. These claims are addressed to the manner in which the control database object specifies the set of files containing the objects and the objects in the set of files. The information specifying the set of files is shown at 521 in FIG. 5 and in detail in 1401 in FIG. 14. The FILE row is discussed at page 26, lines 1-12. The structure of the set of files is shown in overview in FIG. 3, described at page 8, line 34 - page 9, line 13 and in detail in FIG. 9, described at page 11, line 30 - page 12, line 17. The rows in the control database object which represent objects being transferred are shown in overview at 507 in FIG. 5, described at page 11, lines 5-18, and in detail at 507A, B, and C in FIGs. 10-12; the detailed description of the object row may be found at page 22, line 1- page 23, line 22. The "template" of claim 23 is embodied in the WILDCARD file row, shown at 1409 in FIG. 14 and is described at page 26, lines 15-19.

20

25

30

The claim of group H

Claim 24 is directed to an export operation in which the database system which is the source of the objects being exported is a remote database system and the dump files for the export operation are made in the local database system. FIG. 7 has a flowchart for export generally; the export operation is described generally beginning at page 14, line 20, and export in which the source of the objects may be a remote database is described at page 15, lines 21-32.

The claim of group I

Claim 25 is directed to an import operation in which the source of the items being imported is a dump file set. Fig. 8 has a flowchart for this type of import operation and the operation is described in detail at page 16, lines 1-27.

The claims of group J

Claims 26-35 are directed to the control interface for the transfer mechanism. The claims are dependent from claim 1. FIG. 21 provides an overview of the control interface; it is explained in detail at page 29, line 10-page 35, line 16.

The claims of group K

Claims 44-60 are directed to the operation of the transfer mechanism. FIG. 4 is a state machine for the transfer mechanism; it is explained at page 19, line 24-page 21, line 17. FIG. 6 is an overview of the operation of the transfer mechanism. It is explained at page 12, line 20-page 14, line 16.

The claims of group L

Claims 36-38 are dependent from claim 36 and are directed to set of files 323. Claim 63 is a Beauregard claim dependent from claim 36. FIG. 3 provides an overview of the set of files and is described at page 8, line 34-page 9, line 14. FIG. 9 provides details of the set of files and is described at page 11, line 31-page 12, line 17.

30

(6) Grounds of rejection to be reviewed on appeal

The grounds of rejection to be reviewed on appeal are the following:

- the rejection of claims 1,3-15, 23-36, 43, 50-51, 53-55, 58-63, and 65 under 35 U.S.C. 103(a) as obvious over *Oracle 9i Database Documentation* (Release 2 [9.2], March 2002, (hereinafter "Oracle"), in view of U.S. Patent 6,993,529, Basko, et al.,
 5 *Importing data using metadata*, having a filing date of June 1, 2001 and an issue date of Jan. 31, 2006 (hereinafter "Basko") and
- the rejection of claims 16-22, 37-38, 44-49, and 56-57 under 35 U.S.C.103(a) as obvious over Oracle and Basko in view of U.S. Patent 6,032,159, Rivlin, *Apparatus and method for annotated bulk copying [of] files to a database*, issued Feb. 29, 2000
 10 (hereinafter "Rivlin").

(7) Argument

In general

15 As set forth at MPEP 2142, in order to reject a claim under 35 U.S.C. 103(a), Examiner must make a *prima facie* case which has the following elements:

- First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine the reference teachings.
- 20 • Second, there must be a reasonable expectation of success.
- Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. (MPEP 2142, Rev. 5, Aug 2006, p. 2100-125, col. 2)

In the following, Applicants will demonstrate that the Oracle and Basko references do not show all of the limitations of any of the claims and that Examiner has consequently failed
 25 to make his *prima facie* case with regard to all of the claims.

What Applicants are claiming

As indicated above, each of independent claims 1 and 43 contains the limitations which are relevant to the present discussion. Because that is the case, the following argument
 30 regarding claim 1 applies equally to claim 43, as well as to the dependent Beauregard claims 62 and 65.

The embodiment of the invention: FIGs. 3 and 5

The embodiment of the invention is shown in overview in FIG. 3 and described in overview at page 7, line 30 through page 10, line 2. As there described, salient features of the embodiment are the following, described at page 8, line 34-page 9, line 14:

- 5 • *database objects* are "objects that can be manipulated by DBMS programs" (page 8, lines 8 and 9).
- the embodiment exports database objects from a database and imports them into a database (page 8, lines 28-33).
- The export and import operations are controlled by master table 321, which is a
10 database object that represents the import or export job. Master table 321 has the following functions:
 - it determines what objects are imported or exported, how they are imported or exported, and what operations are performed on them in the course of import or export;
 - 15 – it contains information about the current status of the import or export job and permits stopping and restarting a job.
 - it permits a job to continue while the user for whom the job is being performed is detached from the database system.

Master table 321 is shown in detailed overview in FIG. 5 and described at page 10, line 5
20 through page 11, line 29. A salient feature of master table 321 that is described there is that the objects being imported or exported are represented by rows in the master table (page 11, lines 5-19).

The rejection of the claims of group A

25 Applicants' claim 1

Applicants' claim 1 is straightforward. It reads as follows:

1. Apparatus in a database management system for performing a job which transfers a set of database objects into or out of the database management system, the apparatus comprising:
 - 30 a transfer mechanism that transfers database objects; and
 - a queryable control database object that represents the job and specifies the set of objects,

the transfer mechanism operating under control of the control database object to transfer the objects in the set.

The claim is directed to apparatus for transferring a set of database objects into or out of a database management system. The apparatus has the salient features set forth in the foregoing discussions of FIGs. 3 and 5: the job performed by the apparatus "transfers a set of database objects into or out of the database management system; the apparatus includes "a queryable control database object that represents the job and specifies the set of objects" (master table 321); and the apparatus "operates under control of the control database objects to transfer objects in the set".

The rejection of claim 1 under 35 U.S.C. 103

At pp. 4 and 5 of Examiner's final rejection of 10/18/2006, Examiner rejects Claim 1 on the combination of the Oracle 9i SQL Reference and Basko. He cites the following in the Oracle 9i SQL Reference for the claim's "transfer mechanism":

Oracle automatically provides the low-level infrastructure services needed for input-output, heterogeneous client-side access for new datatypes, and optimizations for data transfers between the application and the database. (Oracle 9 SQL Reference, p. 2-40)

For the "control database object that represents the job and specifies the set of objects, Examiner cites the description of the SQL CREATE PROCEDURE in the SQL reference:

Use the CREATE PROCEDURE statement to create a standalone stored procedure or a call specification. A **procedure** is a group of PL/SQL statements that you can call by name. A **call specification ("call spec")** declares a Java method or a third-generation language (3GL) routine so that it can be called from SQL and PL/SQL. The call spec tells Oracle which Java method to invoke when a call is made. It also tells Oracle what type conversions to make for the arguments and return value. Stored procedures offer advantages in the areas of development, integrity, security, performance, and memory allocation. (Oracle 9 SQL Reference, p. 14-64)

Examiner admits that the CREATE PROCEDURE statement is not queryable, but finds that limitation at col. 10, lines 31-59 of Basko. What the cited location describes is a

mechanism for aggregating data from tables in data warehouse 107 into an aggregation table which would also generally be stored in the data warehouse. See the discussion of FIG. 1 at col. 3, lines 55-61. As described at col. 10, lines 31-59, the mechanism is aggregation metadata that describes the sources of the data to be aggregated in the
 5 aggregation table and how the data is to be aggregated. The aggregation metadata is contained in a set of tables that are described as follows:

The meta_Aggregation table contains an entry for each aggregation that is defined. The meta_AggregationFact table contains for each aggregation an entry for each fact table that is used to generate that aggregation. For
 10 example, a single aggregation may be generated from the data of five different fact tables in which case the meta_AggregationFact table would have five entries for that aggregation. The meta_AggregationMeasure table contains for each entry of the meta_AggregationFact table an entry for each measure of the aggregation that is generated from that fact table.
 15 The meta_FactMeasure table has for each fact table an entry for each measure that is generated from the fact table and specifies how that measure is to be generated. The meta_AggregationProperty table specifies the aggregating property for an aggregation. The meta_ReportMeasure table specifies the measure names to be included in the report for the
 20 aggregation. The meta_Fact table contains an entry for each fact table that is used in an aggregation.

Beginning with the claim's "queryable control database object", CREATE PROCEDURE is an SQL statement and an SQL statement is of course not a database object at all. See in
 25 this regard the list of database objects at page 2-102 of the SQL Reference. The procedure created by CREATE PROCEDURE also cannot be the control database object. First, it is not queryable; second, CREATE PROCEDURE only establishes that procedures can be created in database systems; it discloses nothing whatever about particular procedures.

30 The tables containing the aggregation metadata come closest to the claimed "queryable control database object". They are queryable, they do represent a job, namely that of aggregating specific data from specific tables into the aggregation table, and the aggregation engine 108 operates under the control of the aggregation metadata to produce
 35 the aggregation table. The problem is that they do not represent a job which "transfers a set of database objects into or out of the database management system", as required by

claim 1. First, aggregation is not a *transfer* of objects; second, the result of the aggregation is a table in a database management system, and consequently, the database objects are not transferred *into or out of* the database management system. Because the aggregation metadata does not represent a job which "transfers a set of database objects
 5 into or out of the database management system", the "transfer mechanism" cannot operate under control of the metadata tables to "transfer the objects in the set [into or out of the database system], as also required by claim 1.

Because there is nothing disclosed in the Oracle SPL Reference Manual or in Basko that
 10 corresponds to Applicants' queryable control database object or to its interaction with the transfer mechanism, the combined references do not disclose all of the limitations of claim 1 and Examiner has not made his *prima facie* case of obviousness. As will be readily apparent to the Board, the Examiner has also not made his *prima facie* case with regard to claim 43. Further, because claims 1 and 43 are not rendered obvious over the
 15 reference, neither are the claims that are dependent from claims 1 and 43 rendered obvious.

The rejection of the claims of group B

Claims 3 and 4 are directed to the feature of the control database object that the control
 20 database object specifies an order in which the transfer mechanism transfers the objects in the set. Examiner rejects these claims on the basis of the SQL ORDER BY clause and the SQL VSIZE function. One problem with the rejection is that constructs in the SQL language are not data base objects and consequently do not disclose a "control database object" having the added limitations set forth in claims 3 and 4. Another problem is that
 25 these SQL constructs determine the behavior of SELECT, which makes a set of rows in the database system from the table fields specified in the SELECT statement and does not control the transfer of a set of database objects into or out of the database, as required for the control database object. See in this regard the discussion of SELECT at page 8-2 of the SQL Reference:

30 The list of expressions that appears after the SELECT keyword and before the FROM clause is called the **select list**. Within the select list, you specify one or more columns in the set of rows you want Oracle to return

from one or more tables, views, or materialized views. The number of columns, as well as their datatype and length, are determined by the elements of the select list.

5 The SQL ORDER BY clause determines the order of the rows in the set of rows returned by the SELECT statement and the SQL VSIZE function is a function which returns the length in bytes of an expression. Like many other SQL functions, VSIZE can be used by ORDER BY to determine the order of the rows.

10 Because the references do not disclose a control database object which "specifies an order in which the transfer mechanism transfers the objects in the set" or one in which the order "orders the objects in the set by size", Examiner has not made his *prima facie* case of obviousness with regard to claims 3 and 4.

15 *The rejection of the C claim*

Claim 5 is rejected on the basis of the SQL WHERE clause. The arguments made in the discussion of the B claims with regard to the use of SQL constructs as a basis of rejection apply here as well, and Examiner has not made his *prima facie* case of obviousness with regard to claim 5.

20

The rejection of the D claims

Claims 6-8 are rejected on the basis of SQL constructs that perform operations on record fields returned by a SELECT statement. The arguments made in the discussion of the B claims with regard to the use of SQL constructs as a basis of rejection apply here, too, and Examiner has not made his *prima facie* case of obviousness with regard to claims 6-8.

25

The rejection of the E claims

Claims 9-11 are directed to the features of the invention that are related to the job status information which is maintained in the control database object. Examiner refers Applicants to the Oracle 9i User-managed Backup and Recovery Guide for these features. A general problem with this rejection is that backup and recovery does not

30

involve "transfer[ing] a set of database objects into or out of the database management system"; as set forth at 1-2 of the User Managed Backup and Recovery Guide, what it involves is:

5 The basic user-managed backup strategy is to make periodic backups of datafiles and archived logs with operating system commands.

 The basic user-managed procedure for recovering from a media failure is as follows:

1. Restore database file backups with operating system commands.
- 10 2. Recover restored datafiles with the SQL*Plus RECOVER statement.

Neither the datafiles nor the archived logs are database objects as that term is used in the current context; moreover, backups do not "transfer a set of database objects into ... the database system" and recoveries do not "transfer a set of database objects ... out of the database system" as required by Applicants' claims.

As for the specifically-cited locations in the Oracle 9i User-managed backup and recovery guide, Examiner cites page 2-3 in his rejection of claim 9 and 10. This location discloses how to use the system V\$BACKUP view to determine whether a datafile is included in a tablespace that is in backup mode. The view does not include a "specification of a status of the job" where the job is transferring database objects into or out of the database management system.

Page 4-34, cited in the rejection of claim 11, discloses that a canceled recovery may be resumed at the point where it was canceled; a recovery is of course not a transfer of database objects into or out of the database management system and there is further no disclosure of the mechanism used to resume the canceled recovery. Consequently, Examiner has not made his *prima facie* case of obviousness with regard to claims 9-11.

30 *The rejection of the F claims*

Claims 12-13 are directed to using the control database object to specify a remote database management system as a source of the set of objects being transferred. In these rejections, Examiner again employs the SQL SELECT statement, which, as pointed out

above, is not a database object and does not transfer database objects into or out of the database management system. Here, what is cited at page 17-48 is a table of privileges which indicates that a SELECT privilege is available on an external table. External tables themselves are defined as follows at page 15-33 of the SQL reference manual:

5 Use the `external_table_clause` to create an external table, which is a read-only table whose metadata is stored in the database but whose data is stored outside database. External tables let you query data without first loading it into the database, among other capabilities.

10 The external source of the data is explained at 15-35 as follows:

 The `LOCATION` clause lets you specify one or more external data sources. Usually the `location_specifier` is a file, but it need not be. Oracle does not interpret this clause. It is up to the access driver to interpret this information in the context of the external data.

15

As is clear from this citation, the data source for an external table is not a remote database management system and the data source for the external table does not contain database objects. What it contains is data that the database system *interprets as* database objects.

20 Examiner's rejection of claim 13 rests on a misunderstanding of the claim. The antecedent of "the database management system" is "a database management system" in claim 1, line 1, not the "remote database management system" of claim 12, line 1. Since that is the case, Examiner's citation of output to a "remote output table", i.e., one contained in another database management system, is simply irrelevant. Further, as
25 already pointed out, SELECT does not "transfer[] a set of database objects into or out of the database management system".

Examiner has thus failed to make his *prima facie* case of obviousness for claims 12 and 13.

30

The rejections of the G claims

Claims 14 and 15 are addressed to the version of the technique in which the source or destination for the data objects being transferred is dumpfiles 323. In his rejection, Examiner refers to the Oracle 9i SQL Reference's disclosure of external tables which can

be accessed via a SELECT statement. As already pointed out, the SELECT statement is not a database object, and consequently cannot be the "control data base object; moreover, the data source for an external table does not contain database objects and consequently cannot be the "set of files of the claim". See the rejection of claim 12 in
5 this regard. Finally, external tables are read only; consequently, they cannot be a "destination of the set of database objects", as required by the claim. Examiner has consequently not established his *prima facie* case with regard to claim 14 or with regard to any of the claims dependent from that claim.

10 As concerns claim 15, Examiner refers Applicants to stored procedures. As already pointed out, a stored procedure is not queryable and consequently cannot be the copy of the control database object. Further, there is no disclosure whatever in the references concerning a "set of files" that is the result of a job "which transfers a set of database objects" and which "includes a copy of the control database object for the job". For these
15 reasons, Examiner has not established his *prima facie* case with regard to claim 15.

Claims 16-22 are directed to various aspects of the manner in which the control data object is a table database object (see claim 1) that employs table rows to represent the database objects being transferred. The claims are rejected on the basis of the
20 combination of Oracle, Basko, and Rivlin. Rivlin discloses a system in which files are transferred by a bulk copy process to a database system. A "table" in a file server which is not a database table, but rather an object defined outside the database system, is used to keep track of whether a file has been selected for transfer and whether the transfer for a selected file was completed. The "table" has a row for each file to be transferred. The
25 combination of Oracle, Basko, and Rivlin consequently does not disclose the table control data object that "includes rows representing objects belonging to the set of database objects" set forth in claim 16 and Examiner has not made his *prima facie* case with regard to the claim.

30 Claims 17-20 and 22 set forth further limitations concerning the rows in the control database object which represent objects being transferred and for the reasons set forth

above, Examiner has not made his *prima facie* case of obviousness with regard to these claims.

Claim 21 sets forth that the "set of files" of claim 14 includes a copy of the control database object with a row for each database object contained in the set of files and sets forth how the control database object used to import objects into the database management system contains a copy of the rows representing the objects from the copy of the control database object in the file set. As already pointed out with regard to claim 15, there is no disclosure whatever in the references concerning a "set of files" that is the result of a job "which transfers a set of database objects" and which "includes a copy of the control database object for the job", and as one would expect from that fact, no disclosure of the fact that the control database object which controls the transfer includes rows for the objects that are copied from the copy of the control database. Examiner has consequently not made his *prima facie* case with regard to claim 21.

As concerns claim 23, the claimed template is used to add files to the "set of files"; Examiner bases his rejection on the syntax used to create logfiles described at page 13-28 of the Oracle 9i SQL Reference. A logfile is a file that contains a history of transactions on a set of database objects; it is not a set of files that contain "database objects" (claim 14); further, as far as can be determined from the Oracle documentation concerning logfiles, templates cannot be used to specify logfiles. See 7-43 and 7-44 of the Oracle 9i SQL Reference. Examiner has thus not made his *prima facie* case with regard to this claim.

Claims 60 and 61 correspond to claims 16 and 20 except that they are dependent from claim 1 instead of claim 16. Examiner has not made his *prima facie* case for these claims for the reasons indicated in the discussion of claims 16 and 20.

Group H

This group contains claim 24; claim 24 is like claim 12 except that it adds the set of files as a destination for the transfer; the argumentation with regard to claim 12 thus applies

here, as well as the argumentation from claim 14 concerning the "set of files". By either argument, the Examiner has not made his *prima facie* case.

Group I

5 This group contains claim 25; Examiner rejects the claim on the basis of datafiles and references page 9-29 of the SQL reference. Applicants could find no reference to datafiles at that location. Datafiles are, however, defined in the *Glossary* of the document, Oracle 9i Database Concepts, as follows:

data file

10 A physical operating system file on disk that was created by Oracle and contains data structures such as tables and indexes. A data file can only belong to one database.

In terms of the patent application's *Detailed Description*, the datafiles contain the data structures for database 317 in FIG. 3; as shown in that figure, while database 317 and dump files 323 are both contained in DBMS 301's persistent storage 315, database 317 and dump files 323 are separate entities. Consequently, the datafiles making up database 317 cannot be taken to be dumpfiles and Examiner has again failed to make his *prima facie* case.

20

Rejection of the claims of group J

These claims include claims 26-35 dependent from claim 1. The claims are directed to the API (application programmer's interface) for the apparatus of claim 1. Examiner rejects claims 26, 27, 29, and 30 of the SQL*PLUS prompt set forth at page 2-2 of the Oracle 9i User-managed Backup and Recovery Guide. As set forth in the *Glossary* of the Oracle 9i Database Concepts document, SQL*PLUS is an

25

Oracle tool used to execute SQL statements against an Oracle database.
Oracle SQL includes many extensions to the ANSI/ISO standard SQL language.

30

SQL*PLUS is thus an interactive interface which uses SQL to interact with a *database system*, not "an interface whereby an entity that uses the transfer mechanism may interact with the job" where the job "transfers a set of database objects into or out of the database

management system", as set forth in claim 26 by virtue of its dependency from claim 1. It will be immediately apparent from a perusal of FIG. 21 that the API for the apparatus of claim 1 does not consist of SQL statements. Since that is the case, Examiner has not made his *prima facie* case with regard to claims 26, 27, 29, and 30.

5

Concerning claim 28, Examiner refers to the mechanism in the database management system for resolving links to objects in remote database systems. This, of course, has nothing whatever to do with claim 28's "the entity may use the interface [whereby an entity that uses the transfer mechanism may interact with the job]" and Examiner has not made his *prima facie* case with regard to claim 28.

10

As regards claim 31, see the discussion of claim 10 above; as regards claim 33, see the discussion of claim 11 above; as regards claims 34 and 35, see the discussion of claim 26 above.

15

Rejection of the claims of group K

Group K includes claims 44-60. These claims are dependent from claim 43 and are directed to the operations which may be carried out in performing the job. Claims 44-48 and 56-57 are rejected on the basis of the combination of Oracle, Basko, and Rivlin.

20

Claims 44, 45, 47, 49, 56, and 57 have to do with attachment to or detachment from a job. In rejecting all of these claims, Examiner refers Applicants to col. 7, lines 25-58 of Rivlin, which discloses that file table 34 can be used in case of failure to determine what information has been transferred to the database (col. 7, lines 53-55). Applicants respectfully submit that this has nothing whatever to do with the aspects of attachment and detachment set forth in claims 44, 45, 47, 49, 56, and 57. Examiner consequently has not made his *prima facie* case with regard to these claims.

25

With regard to claim 46, Examiner refers Applicants to pp. 7-49 through 7-51 of the Oracle 9i SQL reference. The cited location refers to the DDL parallel clause, which lets one set degrees of parallelism for database objects in the database system. What is

30

claimed, however, is "specifying a degree of parallelism with which the objects may be transferred [into or out of the database management system]. The cited location does not disclose parallelism in this context and consequently, Examiner has not made his *prima facie* case.

5

With regard to claim 48, see the discussion of claim 27 above. As for claim 50, see the discussion in the argument for claim 1 as to why the CREATE PROCEDURE statement cannot be taken to create the control database object.

10 The rejection of claim 51 is based on the FROM clause of the SELECT statement. The FROM clause identifies the database system table from which the SELECT statement will select data to produce another table in the database system. The SELECT statement is of course not a control database object. Further, as disclosed at page 32, lines 3-7, the source specified in claim 51 for an import job is not a local database system table, but
 15 rather a remote database management system or a dump file; the source for an export job may be a remote database management system or the local database management system; with export the destination will be a dump file in all cases. The result of a SELECT statement is of course a table in the database system, not a dumpfile. Consequently, Examiner has not made his *prima facie* case with regard to claim 51.

20

The rejections of claims 53-55 are based on various aspects of the SELECT statement. As previously pointed out, SELECT is not a database object, as required by the claims. Moreover, in claim 55, whatever parameters may apply to a SELECT statement are not parameters for a *job* as that term is defined in Applicants' Specification and claims.
 25 Again, no *prima facie* case has been made for the rejection of these claims. The arguments made with regard to claim 55 also apply to claims 58 and 59.

Rejection of the claims of group L

Group L includes independent claim 36, dependent claims 37 and 38, and dependent
 30 Beauregard claim 63. These claims are directed to the invention embodied in dump files 323. Independent claim 36 contains the limitations of independent claims 1 and 43 and is

patentable over the Oracle and Basko references for the same reasons that claims 1 and 43 are patentable; further limitations that are not shown in Oracle and Basko are the set of files and the fact that the queryable control database object is contained in the set of files and specifies the locations of database objects in the files and the order in which they are to be transferred. Examiner rejects claim 36 on the basis of the disclosure of external tables in the Oracle 9i SQL reference and Basko's aggregation metadata. As already explained with regard to claim 12, external tables do not contain database objects; further, Basko's aggregation metadata consists of database tables in a database management system. They are not "contained in a file belonging to the set of files" as is the queryable control database object of claim 36. Examiner has thus not established his *prima facie* case for the claim or for the claims dependent from the claim.

Claim 37 is rejected on the basis of Oracle, Basko, and Rivlin. Claim 37 sets forth that the file includes metadata for a type of database objects and database objects belonging to the type and that the order specified in the control database object "determines that the metadata is processed before the database objects that belong to the type defined by the metadata". Examiner finds the metadata at col. 3, lines 9-42 of Rivlin. What is described there is a "file attribute table" which describes each file and a set of predetermined attributes associated with each of the files. The file attribute table is then used in bulk loading the files into a database system. The first problem with this rejection is that the claim requires that the metadata define "a type of database objects"; here, the metadata describes files; further, the "file attribute table" is not included with the set of files being bulk loaded; additionally, the only thing that is ordered by the file attribute file is the files being processed; finally, making the "file attribute table" is independent of the bulk loading; the bulk loader does not first process the file attribute table and then process the database objects, all as required by the claim. Examiner has thus not established his *prima facie* case for the claim.

Claim 38 is addressed to the headers used in the dump files. The claim is apparently rejected on the basis of pages 7-8 and 7-9 of the Oracle 9i recovery manager's user guide, which discloses nothing beyond that when copies of backups are checked, their file

headers are checked. This of course says nothing about what is set forth in the claim. Again, no *prima facie* case has been established.

Conclusion

5 In the foregoing, Applicant has complied with the requirements of 37 C.F.R. 41.37 with regard to his brief and has demonstrated in the brief that examiner has failed to establish a *prima facie* case of obviousness with regard to *any* of his rejections under 35 U.S.C. 103. That being the case, the rejections cannot stand and Applicant respectfully requests that the Board of Appeals reverse the examiner with regard to all of his rejections and remand
10 the application to the examiner for further processing as indicated by the reversals. The required fee of \$500 accompanies this brief; should any other fees be required, please charge them to Deposit Account number 501315

Respectfully submitted,

15

/Gordon E. Nelson/

Attorney of record,

Gordon E. Nelson

20

57 Central St., P.O. Box 782

Rowley, MA, 01969,

Registration number 30,093

Voice: (978) 948-7632

Fax: (866) 723-0359

25

4/19/2007

Date

(8) Appendix of claims

1 **1.** Apparatus in a database management system for performing a job which transfers a set of
2 database objects into or out of the database management system, the apparatus comprising:
3 a transfer mechanism that transfers database objects; and
4 a queryable control database object that represents the job and specifies the set of
5 objects,
6 the transfer mechanism operating under control of the control database object to transfer the
7 objects in the set.

1 **2. (canceled)**

1 **3.** The apparatus set forth in claim 1 wherein:
2 the control database object further specifies an order in which the transfer mechanism
3 transfers the objects in the set.

1 **4.** The apparatus set forth in claim 3 wherein:
2 the order orders the objects in the set by size.

1 **5.** The apparatus set forth in claim 1 wherein:
2 the control database object includes a filter that further specifies the set of objects.

1 **6.** The apparatus set forth in claim 1 wherein:
2 the transfer mechanism further performs an operation on one or more objects belonging to
3 the set; and
4 the control database object includes a specification of the operation.

1 **7.** The apparatus set forth in claim 6 wherein:
2 the operation is an operation that transforms the object.

1 **8.** The apparatus set forth in claim 7 wherein:

2 the operation is an operation that remaps a name in the object to a different name.

1 **9.** The apparatus set forth in claim 1 wherein:

2 the control database object includes a specification of a status of the job; and
3 the transfer mechanism updates the status in the specification during the transfer.

1 **10.** The apparatus set forth in claim 9 wherein:

2 the control database object is queryable to obtain a current status of the job from the
3 specification of the status.

1 **11.** The apparatus set forth in claim 9 wherein:

2 the transfer mechanism employs the specification of the status of the job to restart the job
3 after the job has been stopped.

1 **12.** The apparatus set forth in claim 1 wherein:

2 the control database object specifies a remote database management system as a source of
3 the set of objects; and
4 the transfer mechanism fetches the set of objects from the remote database management
5 system.

1 **13.** The apparatus set forth in claim 12 wherein:

2 the control database object specifies the database management system as a destination of
3 the set of database objects; and
4 the transfer mechanism further fetches the set of database objects into the database
5 management system.

1 **14.** The apparatus set forth in claim 1 wherein:

2 the control database object specifies a set of files in the database system as a source or
3 destination of the set of database objects.

1 **15.** The apparatus set forth in claim 14 wherein:

2 when the set of files is the source of the set of database objects, the set of files is the result
3 of a job and includes a copy of the control database object for the job.

1 **16.** The apparatus set forth in claim 14 wherein:

2 the control database object is a table and includes rows representing objects belonging to
3 the set of database objects.

1 **17.** The apparatus set forth in claim 16 wherein:

2 each row representing an object belonging to the set includes a specification of an order in
3 which the object represented by the row is to be transferred from the set of files relative to other
4 objects belonging to the set.

1 **18.** The apparatus set forth in claim 16 wherein:

2 when the set of files is the destination of the set of database objects, there is a row
3 representing each object that has been transferred to the set of files.

1 **19.** The apparatus set forth in claim 16 wherein:

2 when the set of files is the source of the set of database objects, there is a row representing
3 each object which is to be transferred into the database management system.

1 **20.** The apparatus set forth in claim 19 wherein:

2 the row representing a particular object includes a field whose value specifies an order in
3 which the object is to be transferred relative to the other objects.

1 **21.** The apparatus set forth in claim 16 wherein:

2 the set of files is the result of a job and includes a copy of the control database object for
3 the job, the copy having a row for each database object contained in the set of files; and

4 when the transfer mechanism is transferring the objects belonging to the set of objects
5 from the set of files into the database management system, the control database object contains a
6 copy of at least the rows representing the objects from the copy of the control database object in
7 the file set.

1 **22.** The apparatus set forth in claim 16 wherein:

2 the row in the copied rows representing a particular object includes a field whose value
3 specifies an order in which the object is to be transferred relative to the other objects represented by
4 the copied rows.

1 **23.** The apparatus set forth in claim 14 wherein:

2 the control database object further specifies a template whereby the transfer mechanism
3 may add a file to the set of files when required for transferring the objects.

1 **24.** The apparatus set forth in claim 1 wherein:

2 the control database object specifies a remote database management system as a source of
3 the set of objects and a set of files in the database system as a destination therefor; and

4 the transfer mechanism transfers the set of objects from the remote database management
5 system to the set of files.

1 **25.** The apparatus set forth in claim 1 wherein:

2 the control database object specifies a set of files in the database system as a source of the
3 set of objects; and

4 the transfer mechanism transfers the set of objects from the set of files into the database
5 management system

1 **26.** The apparatus set forth in claim 1 wherein:

2 the transfer mechanism further provides an interface whereby an entity that uses the
3 transfer mechanism may interact with the job.

1 **27.** The apparatus set forth in claim 26 wherein:

2 the interface permits the entity to attach to and detach from the job for as long as the job's
3 control database object exists, transfer of the objects by the transfer mechanism being unaffected
4 by detachment of the entity from the job.

1 **28.** The apparatus set forth in claim 26 wherein:

2 the entity may use the interface via a network connection to the database management
3 system.

1 **29.** The apparatus set forth in claim 26 wherein:

2 the interface includes a defining interface whereby the entity may define a portion of the
3 job's control database object.

1 **30.** The apparatus set forth in claim 26 wherein:

2 the interface includes an executing interface whereby the entity may interact with the
3 transfer mechanism from the time the transfer mechanism begins transferring the objects in the set
4 until the job's control database object ceases to exist.

1 **31.** The apparatus set forth in claim 30 wherein:

2 the entity may use the executing interface to obtain a current status of the job from a
3 specification of the status of the job in the control database object.

1 **32.** The apparatus set forth in claim 27 wherein:

2 the entity may use the executing interface to stop performance of the job by the transfer
3 mechanism or the transfer mechanism may stop performance of the job in response to an error.

- 1 **33.** The apparatus set forth in claim 32 wherein:
2 the entity may use the executing interface to restart a stopped job, the transfer mechanism
3 using a specification of the status of the job in the control database object to restart the job.
- 1 **34.** The apparatus set forth in claim 30 wherein:
2 the entity may use the executing interface to affect allocation of resources by the transfer
3 mechanism to the job.
- 1 **35.** The apparatus set forth in claim 34 wherein:
2 the transfer mechanism operates on objects in the set in parallel; and
3 the entity uses the executing interface to specify a maximum degree of parallelism for the
4 job.
- 1 **36.** A set of files for transferring a set of database objects into a database management system,
2 the set of files comprising:
3 at least one file containing the objects belonging to the set thereof; and
4 a queryable control database object contained in a file belonging to the set of files that
5 specifies for each object belonging to the set the location of the object in the set of files and an
6 order in which the database management system transfers the object during the transfer.
- 1 **37.** The set of files set forth in claim 36 wherein:
2 the file further includes metadata that defines a type of database objects and one or more
3 database objects that belong to the type defined by the metadata; and
4 the order determines that the metadata is processed before the database objects that belong
5 to the type defined by the metadata.
- 1 **38.** The set of files set forth in claim 36 further comprising:
2 a header in each file of the set, the header including

an indication the control object is contained in the file and if so, the location of the control object in the file and
an identifier that identifies the file within the set; and
the control object uses the identifier in specifying the location of the object in the set of files.

39. (canceled)

40. (canceled)

41. (canceled)

42. (canceled)

43. A method of performing a job that transfers a set of database objects into or out of a database management system that includes a transfer mechanism that transfers the database objects, the method comprising the steps of:
defining a queryable control database object that represents the job and specifies the set of objects; and
executing the job by causing the transfer mechanism to transfer the set of database objects under control of the control data base object.

44. The method of performing a job set forth in claim 43 further comprising the step performed in either the defining step or the executing step of:
attaching to the job, attachment permitting at least reading and/or modification of the job's control database object.

45. The method of performing a job set forth in claim 44 further comprising the step performed after the step of attaching to the job of:

3 reading the job's control database object to get the job's current status.

1 **46.** The method of performing a job set forth in claim 44 wherein
2 the transfer mechanism transfers the data objects in parallel and
3 the method further comprises the step performed after the step of attaching to the job of:
4 specifying a degree of parallelism with which the objects may be transferred.

1 **47.** The method of performing a job set forth in claim 44 wherein the method further
2 comprises the step performed after the step of attaching to the job of:
3 starting the step of executing the job.

1 **48.** The method of performing a job set forth in claim 44 wherein the method further
2 comprises the step performed after attaching to the job of:
3 stopping the step of executing the job.

1 **49.** The method of performing a job set forth in claim 48 wherein the step of stopping the
2 step of executing the job further comprises the step of:
3 saving job state in the control database object such that the step of executing the job
4 may be restarted from the job state.

1 **50.** The method of performing a job set forth in claim 43 wherein:
2 the step of defining the job includes the step of creating the job's control database
3 object.

1 **51.** The method of performing a job set forth in claim 43 wherein:
2 the step of defining the job includes the step of specifying a source and/or destination
3 for the set of database objects in the job's control database object.

1 **52. (canceled)**

1 **53.** The method of performing a job set forth in claim 43 wherein:

2 the step of defining the job includes the step of specifying a filter in the job's control
3 database object, the filter defining a subset of the specified set of database objects as the set
4 of objects to be transferred in the job.

1 **54.** The method of performing a job set forth in claim 43 wherein:

2 the step of defining the job includes the step of specifying an operation in the job's
3 control database object that is to be performed on one or more objects in the set.

1 **55.** The method of performing a job set forth in claim 43 wherein:

2 the step of defining the job includes the step of defining a parameter for the job in the
3 job's control database object for the job.

1 **56.** The method of performing a job set forth in claim 43 wherein

2 the step of executing the job includes the step performed when the step of executing
3 the job must be stopped of:

4 saving job state in the control database object such that the stopped executing step
5 may be restarted from the job state.

1 **57.** The method of performing a job set forth in claim 56 wherein the step of executing the
2 job includes the step performed when the step of executing the job has been stopped of:

3 using the job state to restart the stopped executing step.

1 **58.** The apparatus set forth in claim 1 wherein:

2 the control database object includes a specification of one or more parameters for the
3 job,

4 the transfer mechanism transferring the objects in the set as specified by the parameter.

1 **59.** The apparatus set forth in claim 58 wherein:

2 the parameter is an estimate only parameter,
3 the transfer mechanism responding thereto by providing an estimate of the space required for
4 the objects in the set without transferring the objects.

1 **60.** The apparatus set forth in claim 1 wherein:
2 the control database object is a table and includes rows representing the objects
3 belonging to the set of database objects.

1 **61.** The apparatus set forth in claim 60 wherein:
2 the row representing a particular object includes a field whose value specifies an
3 order in which the object is to be transferred relative to the other objects.

1 **62.** A data storage device characterized in that:
2 the data storage device contains code which, when executed by a processor,
3 implements the apparatus set forth in claim 1.

1 **63.** A data storage device characterized in that:
2 the data storage device contains the set of files set forth in claim 36.

1 **64. (canceled)**

1 **65.** A data storage device characterized in that:
2 the data storage device contains code which, when executed by a processor,
3 implements the method set forth in claim 43.

4

(9) Evidence appendix

None.

(10) Related proceedings appendix

None.